NAVAL AEROSPACE MEDICAL RESEARCH LABORATORY
NAVAL AIR STATION, PENSACOLA, FL 32508-5700

NAMRL TECHNICAL MEMORANDUM 92-1

A COMPUTER-BASED VISUAL ANALOG SCALE

C.A. DeJohn, M.J. Marr, E.A. Molina
and A.H. McCardie

Reviewed and approved 23 June 92

A. J. MATECZUN, CAPT, MC USN
Commanding Officer

This research was sponsored by the Naval Medical Research and Development Command under work unit 63706N M0096.002 7010.

The views expressed in this report are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

Volunteer subjects were recruited, evaluated, and employed in accordance with the procedures specified in Department of Defense Directive 3216.2 and Secretary of the Navy Instruction 3900.39 series. These instructions are based upon voluntary informed consent and meet or exceed the provisions of prevailing national and international guidelines.

Trade names of materials and/or products of commercial or non-government organizations are cited as needed for precision. These citations do not constitute official endorsement or approval of the use of such commercial materials and/or products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1992	

4. TITLE AND SUBTITLE
A Computer-based Visual Analog Scale

5. FUNDING NUMBERS
63706N M0096.002 7010

6. AUTHOR(S)
C.A. DeJohn, M.J. Marr, E.A. Molina, and A.H. McCardie

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Aerospace Medical Research Laboratory
Bldg. 1953, Naval Air Station
Pensacola, FL 32508-5700

8. PERFORMING ORGANIZATION REPORT NUMBER
NAMRL Technical Memorandum 92-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Medical Research and Development Command
National Naval Medical Center
Building 1, Tower 12
8901 Wisconsin Avenue
Bethesda, MD 20889-5044

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Although substantial literature exists on visual analog scales (VAS), much is directed toward the clinical assessment of pain. As part of a Naval Aerospace Medical Research Laboratory (NAMRL) effort to investigate the chronopharmacokinetics of methamphetamine during simulated sustained flight operations we developed a computerized VAS to determine the intensity of subjective responses associated with the administration of the stimulant. With simple modifications to the program, the questionnaire could be modified and used to test subjects' responses to a number of drugs, with better resolution of response than hand-scored methods allow.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Visual Analog Scale, Stimulant, Drugs, Methamphetamine, Questionnaire	29
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Acknowledgments

We would like to acknowledge the assistance of LT Scott Shappell, MSC, USNR for his guidance and advice in the preparation of this publication. We greatly acknowledge Mr. Scott Meyer and Dr. Tatree Nontasak for reviewing the manuscript. A special thanks is also given to HMCS(SW) R. Ford, HM1 Flowers, ET1 J. Brown, HM2 Foti, HM2 M. Cross, HM3 E. Snell, HM3 D. Hernandez, HN J. Kendall, and Mr. Jon Shelley for their tremendous contributions during the data collection.

OVERVIEW

No computerized visual analog scale (VAS) exists to measure the intensity of drug effects reported by individuals. Visual analog scales may be capable of providing rapid and reliable assessment of the degree of various subjective effects due to pharmacological interventions.

We found the VAS to be a reliable means of rapidly obtaining the intensity of subjective responses to pharmacological intervention with methamphetamine. The VAS has demonstrated improved resolution over hand-scored techniques. In addition, it is user friendly, and the questionnaire can be adapted for other drugs with only minor modifications to the program.

BACKGROUND

Substantial literature exists on visual analog scales (1-5). Much of this literature is focused on clinical pain assessment where they have served as sensitive, reliable, and valid indices of pain sensations (6-8). The patient places a mark on a line between two extremes corresponding to the level of pain perceived. The position of this mark is then measured by the experimenter, and the measurement becomes the score for that patient. Visual analog scales (VAS) have advantages over scales that require a specific response selected from a limited number of possible choices. A VAS offers the potential for increased discrimination in response, without the patient having to read and contemplate several choices. The investigator could conceivably divide the line segment into any desired number of increments, depending on the level of discrimination desired. Pencil-and-paper versions are limited by the length of the line and the ability of the experimenter to discriminate sufficiently small measurements. Of special interest to our needs, however, are a number of studies demonstrating the use of such scales to evaluate clinical and experimental drug effects. In an example that combined both pain and drug effects, Price (7) measured sensory intensity and affective responses to experimental hyperthermic pain with visual analog scales. This investigation showed dose-related changes with morphine in differential responses to the sensory and affective scales. In a study directly comparing a check list versus a VAS to assess antihistamine drug side effects, Lundberg (2) found the VAS to be the more sensitive measure.

In summary, the literature indicates that visual analog scales could provide a quick, sensitive, reliable, and valid assessment of the degree of various subjective effects. As part of a Naval Aerospace Medical Research Laboratory (NAMRL) effort to investigate the chronopharmacokinetics of methamphetamine during simulated sustained flight operations we developed a computerized VAS to determine the intensity of subjective responses associated with the administration of the drug. To our knowledge, this is the first computerized VAS of its kind to be developed.

QUESTIONNAIRE DEVELOPMENT

A list of major subjective effects of methamphetamine was generated from the literature (9-11). The items included physical symptoms such as "dry mouth" and "palpitations," negative feelings such as "anxiety" and "depression," and positive effects such as "euphoria" and "alertness." The resulting questionnaire is shown in Appendix A.

HARDWARE/SOFTWARE

A computerized VAS was constructed to allow for automated data collection. The program was written in GW BASIC Version 3.0 for an IBM compatible 286 microcomputer using MS-DOS Version 3.0. A program listing is contained in Appendix B.

An additional program was developed to print the data from multiple sessions in ASCII format, arranged in the order depicted in Appendix A. This program was compiled in Turbo C++ Version 1.0 for an IBM compatible 286 microcomputer (12). A listing of this program is shown in Appendix C.

PROGRAM OPERATION

To start the program, type GWBASIC or BASICA at the DOS prompt. Typing LOAD "<drive:>\<path>\SYMPTOM" will load the program. To run the program type RUN and press <CR>. The following information will then be displayed on the screen:

SUBJECT #

<T>est or <B>ase

SESSION #

Enter the subject number, session number, and whether the session is part of the base-line training week or the test week, where appropriate. A set of instructions to the subject will then be displayed on the screen. These instructions are contained in Appendix D.

Each administration of the questionnaire presents the block of items in random order on a CRT display. The subject is then asked to indicate the degree of intensity of the item by moving an arrow along a horizontal scale from 0 to 100 using the left or right "arrow" keys on the computer keyboard or other input device. The initial position of the arrow is always in the center of the scale to prevent biasing the subject's responses toward either extreme of the scale. An example of a screen item is shown in Fig. 1. The measurement is recorded when the subject depresses the "enter" key. If a response has been entered in error, the subject can return to the previous item by depressing the "back-space" key. The program records and stores the selected positions as numbers between 0 and 100 to the nearest whole number. Administration time averages approximately 3.5 min for the questionnaire listed in Appendix A.

Due to the delay associated with the computer keypad, control of the arrow movement by this method may not be satisfactory. This problem can be solved by using a joystick, mouse, or other input device. We used a Systems Research Laboratories Mini-Modulus III interfaced to the computer by a Labpak multifunction data acquisition board. When the Mini-Modulus III is used as the input device, the "red" key is used to move the arrow to the right, and the "blue" key is used to move the arrow to the left. The subject can return to the previous item by depressing the "zero" key. Responses are entered by depressing the"tapper" key.

PROGRAM OUTPUT

Results are sent to the printer and to a disk for storage. The program first searches for a hard drive to store the responses. If none is found, the data will be stored on drive A. If drive A is unavailable, the data are stored on drive B. There are no error checks or responses.

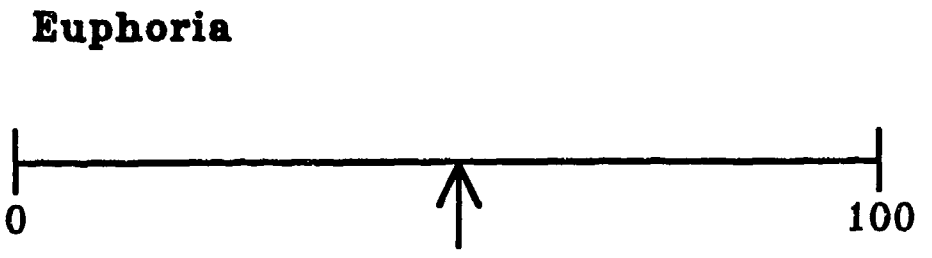

Figure 1. *An example of a screen item.*

DISCUSSION

The VAS appears to be a reliable means of rapidly obtaining the intensity of a subject's responses to pharmacological intervention. In addition, it is user friendly and with minor modifications to the program as detailed in Appendix B, experimenters should be able to adapt the questionnaire to their individual research needs.

The computerized VAS appears to have two major advantages over paper-and-pencil versions of the test. First, it eliminates hand scoring and recording the results and the errors associated with those tasks. Second, it provides greater resolution than hand-scored methods.

REFERENCES

1. Gift, A.G., "Visual Analogue Scales: Measurement of Subjective Phenomena." *Nursing Research*, Vol. 38, pp. 286-288, 1989.

2. Lundberg, P.K., "Assessment of Drugs' Side Effects: Visual Analogue Scale Versus Check-list Format." *Perceptual and Motor Skills*, Vol. 50, pp. 1067-1073, 1980.

3. Menkes, D.B., Howard, R.C., Spears, G.F., and Cairns, E.R., "Salivary THC Following Cannabis Smoking Correlates With Subjective Intoxication and Heart Rate." *Psychopharmacology*, Vol. 103, pp. 277-279, 1991.

4. Newhouse, P.A., Belenky, G., Thomas, M., Thorne, D., Sing, H.C., and Fertig, J., "The effects of d-amphetamine on Arousal, Cognition, and Mood After Prolonged Total Sleep Deprivation." *Neuropsychopharmacology*, Vol. 2, 153-164, 1989.

5. Simpson, M.A., Schoeman, H.S. & Allman, B.J., "A Profile of Buspirone in the Treatment of Anxiety in General Practice Patients." *Current Therapeutic Research*, Vol. 46, pp. 980-992, 1989.

6. Hodgkins, M., Albert, D., & Daltroy, L., "Comparing Patients' and Their Physicians' Assessments of Pain." *Pain*, Vol. 23, pp. 273-77, 1985.

7. Price, D.D. "A Psychophysical Analysis of Morphine Analgesia." *Pain*, Vol. 22, pp. 261-269, 1985.

8. Price, D.D., Rafii, A., Watkins, L.R., and Buckingham, B., "Psychophysical Analysis of Acupuncture Analgesia." *Pain*, Vol. 19, pp. 27-42, 1984.

9. Rall, T.W., "Central Nervous System Stimulants." In A.G. Gilman, L.S. Goodman, T.W. Rall, and F. Murad (Eds.), *The Pharmacological Basis of Therapeutics*, pp. 589-603, MacMillan, New York, NY, 1985.

10. Martin, W.R., Sloan, J.W., Sapira, J.D., and Jasinski, D.R., "Physiologic, Subjective, and Behavioral Effects of Amphetamine, Methamphetamine, Ephedrine, Phenmetrazine, and methylphenidate in Man." *Clinical Pharmacology and Therapeutics*, Vol. 12, pp. 245-258, 1971.

11. Sifton, D.W., Bunnell, C.C., and Pasinski, J.T. (Eds.), *Physicians Desk Reference*, Medical Economics Company, Oradell, NJ, 1990.

12. *Turbo C++*, Borland International Inc., Scotts Valley, CA., 1990.

APPENDIX A

This appendix contains a list of the questions presented during the administration of the drug symptom questionnaire. Each item initially appeared over a 100-mm line with the arrow at the center of the line as shown in Fig. A-1 of the text.

Palpitations (heart fluttering)
Dizziness
Dysphoria (feeling of being ill at ease)
Euphoria (feeling that all is well, a high)
Overstimulation
Insomnia
Restlessness
Tremor (shaking)
Headache
Diarrhea
Constipation
Abdominal cramps
Dry mouth
Unpleasant taste
Urticaria (itching)
Fatigue
Depression
Anxiety
Hostility
Jumpy
Talkative
No appetite
Difficulty urinating
Increased frequency of urination
Sleepiness
Rapid breathing

APPENDIX B

This appendix contains the program that presents the questionnaire, records and prints the responses. It is written in GW BASIC, Version 3.0, for a Z-248 (IBM 286 compatible machine) using MS-DOS Version 3.0.

Introduction. The drug effects questionnaire was designed to present a set or group of drug- related questions to a subject, via a CRT display, for her/(his) rating. As shown in Fig. 1 of the text, the subject is asked to indicate the degree of intensity of the symptom, feeling, or behavior by placing an arrow along a line-segment scale measuring from 0 to 100. The arrow can be moved on the 0-100 scale to indicate the intensity of the particular symptom.

The program was modularly designed so that different questionnaires can be easily adapted. Figure B-1 shows a flow-chart diagram for the program.

Test description. The test consists of five parts:

(a) Initial dialogue.

(b) Questionnaire file retrieval.

(c) Randomization of questionnaire presentation.

(d) Questionnaire presentation and response recording.

(e) Storage and printout of questionnaire and test results.

Initial dialogue. At the start of the test, the program informs the subject, via the CRT display, the purpose of the test, how to provide a response, and gives the subject a chance to practice enter inputing a response.

Questionnaire file retrieval. The program retrieves data from the same directory from which it is running the questionnaire. The questions are contained in a file named "question.sym." The number of questions must match the number of the FOR...LOOP in line number 1450 of the program.

Randomization of questionnaire presentation. The program generates a set of random numbers from 1 to the number of questions. These random numbers are then used as indices to the questions for random presentation.

Questionnaire presentation and response recording. The program presents the questions one at a time, waiting for each individual response from the subject. The start and finish time for presentation of the questionnaire are recorded for later storage.

Storage and printout of questionnaire and test results. The questions, the order of presentation, and the subject's response are stored on a floppy or hard disk as determined by the user.

Questionnaires. The questionnaire was stored in the directory from which the program was retrieved and run.

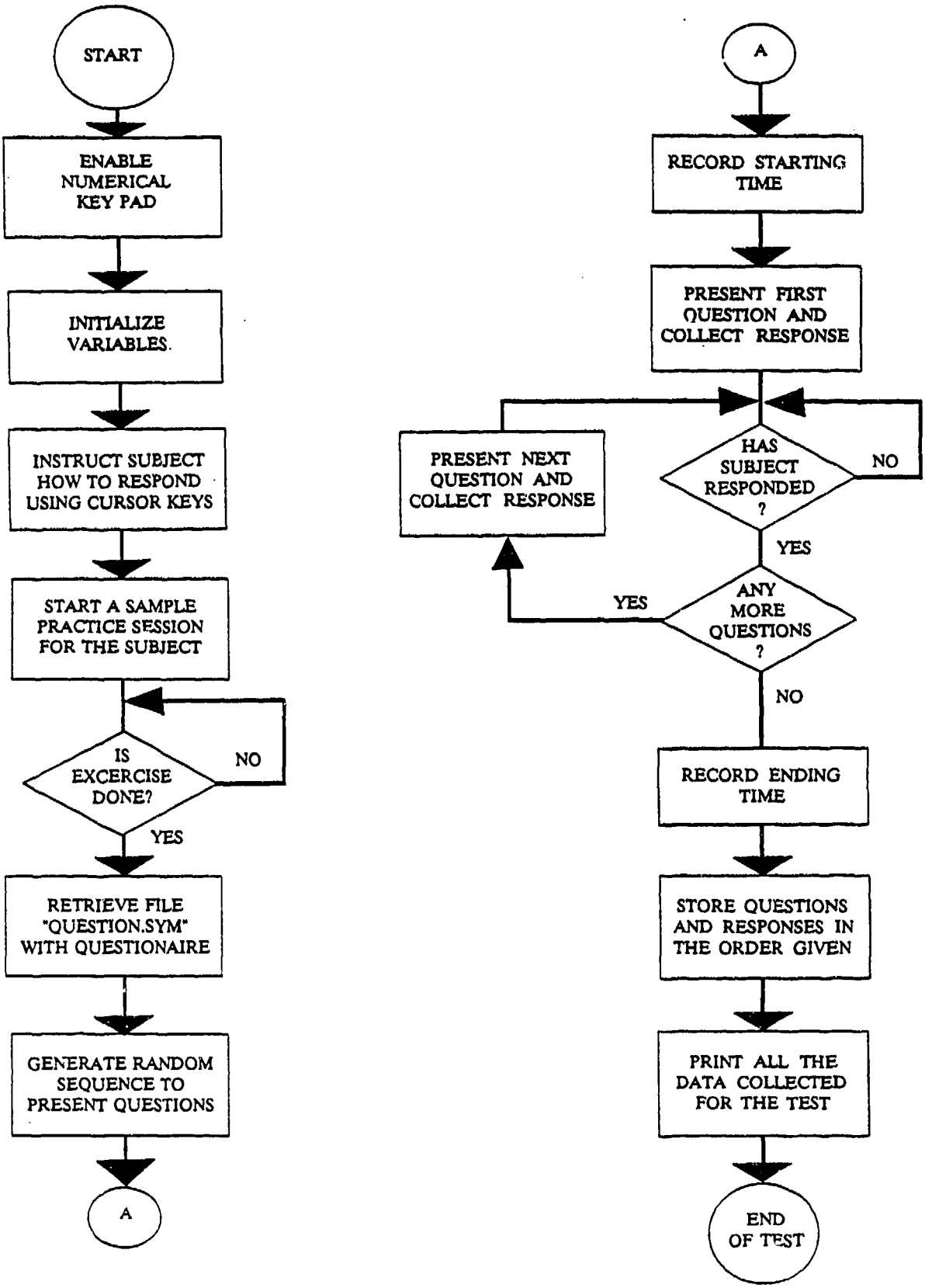

Figure B-1. Flow-chart diagram for drug symptom questionnaire program.

Questionnaires must be created in ASCII format so that each question is terminated with a carriage return. The program reads each question serially as a string assigned to string variable Q$(index). Preparation of the "question.sym" file can be done with WordPerfect, but care must be taken to store it as an ASCII file only. If a comma needs to be presented as part of a question, the entire question needs to be enclosed between quotation marks.

When a new or modified questionnaire is to be incorporated in the drug effects questionnaire program, the following procedure should be followed:

1. Determine the total number of questions in the questionnaire that will be used. Assume the number is NQ.

2. Draft each question, using the same format as those of the questions shown in Appendix A.

3. Use a word processor program (preferably WordPerfect) to write down the questions. Each question must be finished with a carriage return (HRt- in WordPerfect).

4. If the question has a comma throughout the text, the entire question must be enclosed in two quotes. For example:

"Euphoria (feeling that all is well, a high)"

5. Store the questionnaire as an ASCII text file (CTRL-F5 if using WordPerfect 5.1).

6. Make the following changes to the program:

LINE NUMBER	NEW PROGRAM LINE
870	DIM Q$(NQ+3), R(NQ+3), A$, N(NQ+3)
1370	FOR I = 0 TO NQ - 1
1530	S2=NQ-2
1540	IF S1=1 THEN S2=NQ-3
1550	N(NQ-1) = NQ-1
1580	N(I)=INT(RND*(NQ-1))

7. Save the program changes to a file as well as to the printer.

8. Load and run the program to verify operation.

A program listing is shown below:

```
10  'DOCUMENTATION REMARKS
20  '*************************************************************/
30  '*                                                           */
40  '*  The file name of this program is Symptom1.bas            */
50  '*                                                           */
60  '*  RETURN or ENTER key.                                     */
70  '*                                                           */
80  '* Descriptive name: Drug Effects Questionnaire Test         */
90  '*                                                           */
100 '*************************************************************/
110 '*  NAVAL AEROSPACE MEDICAL RESEARCH LABORATORY.             */
120 '*                                                           */
130 '*  DEPARTMENT: Medical Research.                            */
140 '*                                                           */
150 '*  DIVISION    Aviation Performance                         */
160 '*                                                           */
170 '*  AUTHOR(S):  Chuck DeJohn                                 */
180 '*              Jack Marr                                    */
190 '*              Efrain A. Molina                             */
195 '*                                                           */
200 '*                                                           */
210 '*  PROGRAMMER: Efrain A. Molina.                            */
220 '*              Bioengineering Division                      */
230 '*                                                           */
240 '*  DATE:   1 July 1990                                      */
250 '*                                                           */
260 '*  PROJECT:  Effects of Psychopharmacological Counter-      */
261 '*            measures on Performance Decrement During       */
262 '*            Sustained Flight Operations                    */
270 '*                                                           */
280 '*************************************************************/
290 '*  PURPOSE:  To present drug-related questions to a         */
291 '*  subject, using a visual analog scale (VAS), and to       */
292 '*  record and store their responses.                        */
310 '*                                                           */
320 '*  HARDWARE REQUIREMENTS.                                   */
330 '*                                                           */
340 '*    CPU:  Zenith Z-248 PC/AT IBM compatible computer       */
350 '*    MEMORY: minimum of 640K                                */
360 '*    INTERFACES:  UTCPAB standard interfaces.               */
370 '*                                                           */
380 '*                                                           */
390 '*  INPUT:  Subject enters responses via the keyboard        */
400 '*  using the cursor keys (left and right arrows).           */
410 '*                                                           */
420 '*  Name of Output File for data storage, via the key-       */
430 '*  board using the cursor keys (left and right arrows).     */
440 '*                                                           */
450 '*  OUTPUT: Computer presents questions to subject, via      */
460 '*  the CRT, as a horizontal graduated scale from 0 to       */
470 '*  100.                                                     */
480 '*  Test results are sent to the printer and to a disk       */
490 '*  for storage under the name prompted as Output File.      */
500 '*                                                           */
510 '*  ASSUMPTIONS AND LIMITATIONS:  Subject needs to an-       */
520 '*  swer all questions presented by first positioning        */
530 '*  the cursor (right or left) followed by pressing the      */
540 '*  RETURN or ENTER key.                                     */
550 '*                                                           */
560 '*  ERROR CHECKS AND RESPONSES:    None.                     */
570 '*                                                           */
580 '*  ALGORITHM/STRATEGY:  Questions presented via the CRT     */
```

```
590 '*  to the subject are randomly selected by a subroutine*/
600 '*  that generates an array N(i) of random numbers (0   */
610 '*  through 25 or 26 depending on the value of variable */
620 '*  S1 (0 or 1) that can only be set by programming     */
630 '*  change (1 or 0) in line number 1570.                */
640 '*  The questions are then read from an input file named*/
650 '*  "question.sym" into an string array Q$(i).          */
660 '*  Questions are then presented to the subject and a   */
670 '*  response for each question is stored in an array    */
680 '*  R(i).  After all the questions presented have been  */
690 '*  answered the program stores the arrays N(i), Q$(i), */
700 '*  and R(N(i)) in the order indicated in a file named  */
710 '*  by the string variable G$.                          */
720 '*                                                      */
730 '******************************************************/
740 '*  STANDARD OPERATING PROCEDURE.                       */
750 '*                                                      */
760 '*  To be determined by the Nestar/SUSOPS battery.      */
770 '*                                                      */
780 '*                                                      */
790 '*                                                      */
800 '*                                                      */
810******************************************************/
820 REM
830 REM
840 REM To enable the [NUM LOCK] key "ON".
850 KEY OFF
860 GOSUB 2340
870 DIM Q$(30),R(30),A$,N(30)
880 CLS
890 A0=13:Y0=112
900 '******************************************************/
910 '*  Initial dialogue with test subject.  How to respond */
920 '*  using the right and left cursor.  An initial chance */
930 '*  to practice using the cursor.                       */
940 '******************************************************/
950 SCREEN 2
960 A=10
970 PRINT TAB(A) "In this questionnaire, you are asked to rate the degree or
    intensity"
980 REM print
990 PRINT TAB(A) "of a set of symptoms, feelings, or behaviors."
1000 PRINT
1010 PRINT TAB(A) "Each symptom will be presented individually on the screen
     above a line"
1020 REM PRINT
1030 PRINT TAB(A) "segment scale measuring from 0 to 100."
1040 PRINT
1050 PRINT TAB(A) "The arrow can be moved on the 0-100 scale to
     indicate the intensity"
1060 REM PRINT
1070 PRINT TAB(A) "of that particular symptom."
1080 PRINT
1090 PRINT TAB(A) "Press the left or right arrow keys on the keypad until the"
1100 REM PRINT
1110 PRINT TAB(A) "position of the arrow is most appropriate to the intensity"
1120 REM PRINT
1130 PRINT TAB(A) "of the symptom as you are experiencing it NOW."
1140 PRINT:PRINT
1150 PRINT
1160 PRINT
```

```
1170 PRINT TAB(A) "The 0 means no sensation of that symptom
     whatsoever, and"
1180 REM PRINT
1190 PRINT TAB(A) "the 100 means as intense as you could imagine
     experiencing it."
1200 PRINT
1210 PRINT TAB(A) "After you have made your judgement of the symptom
     intensity,"
1220 REM PRINT
1230 PRINT TAB(A) "record your response by pressing the [Enter] key."
1240 PRINT
1250 PRINT TAB(A) "YOU MUST RESPOND TO EACH ITEM.  You can practice now."
1260 GOSUB 1960
1270 CLS
1280 LOCATE 10,5
1290 '*********************************************************/
1300 '*  REM Reading the prompting file that has the ques- .*/
1310 '*  tions. The file name is "question.sym" and should  */
1320 '*  be in the directory from which the program is run- */
1330 '*  ning.  There are a total of 27 questions.          */
1340 '*********************************************************/
1350 LOCATE 10,75
1360 OPEN "I",#1,"question.sym"
1370 FOR I=0 TO 26
1380 INPUT #1, Q$(I)
1390 NEXT I
1400 CLOSE #1
1410 '*********************************************************/
1420 '*  REM Random number sequence generation for          */
1430 '*  presentation of questions.                         */
1440 '*********************************************************/
1450 CLS
1460 A$=TIME$
1470 A%=0
1480 FOR I=1 TO 4 STEP 3
1490 A%=A%+VAL(MID$(A$,I,2))
1500 NEXT I
1510 RANDOMIZE A%
1520 S1 = 0 :REM set s1=1 if question ".. placebo/drug .." is not used
1530 S2=25
1540 IF S1=1 THEN S2 =24
1550 N(26)=26
1560 FOR I=0 TO S2
1570 S3=S2+1
1580 N(I)=INT(RND*26)
1590 IF I=0 THEN GOTO 1630
1600 FOR K=0 TO I-1
1610 IF N(I)=N(K) THEN GOTO 1580
1620 NEXT K
1630 NEXT I
1640 '*********************************************************/
1670 '*  REM Presentation of questionnaire.                 */
1650 '*  Randomized questions are presented in a sequence.  */
1660 '*********************************************************/
1680 S1$=TIME$
1690 Y0=50
1700 A0=5
1710 FOR K=0 TO S3
1720 GOSUB 1960
1730 CLS
1740 NEXT K
1750 S2$=TIME$
```

```
1760 CLS
1770 '*************************************************************/
1780 '*  REM Recording results in file name G$.                   */
1790 '*  The name G$ has the drive path where results are         */
1800 '*  stored.                                                  */
1810 '*  The data is stored in the following sequence:            */
1820 '*  (a) Start time S1$,  (b) Finish time S2$                 */
1830 '*  (c) Number of question asked, (d) Text of question,      */
1840 '*  (e) The numerical response of the scale.                 */
1850 '*************************************************************/
1860 INPUT "Output file's name",G$
1870 OPEN "O",#1,G$,128
1880 PRINT #1,S1$,S2$
1890 FOR I=0 TO S2
1900 PRINT #1, N(I);Q$(N(I));R(N(I))
1910 NEXT I
1920 CLOSE #1
1930 GOSUB 2390
1940 GOSUB 2420
1950 END
1960 '*************************************************************/
1970 '*                                                           */
1980 '*  End of this program.                                     */
1990 '*                                                           */
2000 '*************************************************************/
2010 '*************************************************************/
2020 '*  REM Subroutine used to present the questions and         */
2030 '*  collect the responses.                                   */
2040 '*************************************************************/
2050 REM SCREEN 2
2060 X0=90:X1=590
2070 REM Y0=112
2080 X2=340
2090 IF D1=0 THEN LOCATE 1,15:PRINT Q$(N(K))
2100 IF N(K)=26 THEN LOCATE 3,15
2110 IF N(K)=26 THEN PRINT "Set the arrow to indicate the confidence in your
     decision"
2120 LOCATE A0,12
2130 IF N(K)<>26 THEN PRINT "0"
2140 IF N(K)=26 THEN PRINT "PLACEBO"
2150 LOCATE A0,74
2160 IF N(K)<>26 THEN PRINT "10"
2170 IF N(K)=26 THEN PRINT "DRUG"
2180 LINE (X0,Y0)-(X1,Y0)       : REM Horizontal line
2190 LINE (X1,Y0-5)-(X1,Y0+5)   : REM right-hand vertical mark
2200 LINE (X0,Y0-5)-(X0,Y0+5)   : REM left-hand vertical mark
2210 LINE (X2,Y0)-(X2-5,Y0+5)   : REM left-right line arrow
2220 LINE (X2,Y0)-(X2,Y0+10)    : REM center line arrow
2230 LINE (X2,Y0)-(X2+5,Y0+5)   : REM right-left line arrow
2240 REM LINE (590,50)-(585,55)
2250 REM LINE (590,50)-(595,55)
2260 XB=X2
2270 A$=INKEY$
2280 IF A$=CHR$(13) GOTO 2400
2290 IF A$="" THEN 2270
2300 IF A$="4" THEN X2=X2-5
2310 IF A$= "4" AND X2<90 THEN X2=90
2320 IF A$= "6" THEN X2=X2+5
2330 IF A$= "6" AND X2>590 THEN X2=590
2340 FOR I= 1 TO 10
2350 IF I<6 THEN PRESET (XB-I,Y0+I),0
2360 IF I <6 THEN PRESET (XB+I,Y0-I),0
```

```
2370 PRESET (XB,Y0+I),0
2380 NEXT I
2390 GOTO 2200
2400 R(N(K))= (X2-90)/5
2410 RETURN
2420 '*********************************************************/
2430 '*  REM Routine to turn the [NUM LOCK] key "ON"          */
2440 '*********************************************************/
2450 DEF SEG=&H0
2460 A = PEEK(&H417): B = PEEK(&H418)
2470 POKE &H417,&H20: POKE &H418,&H20
2480 RETURN
2490 '*********************************************************/
2500 '*  REM To turn [NUM LOCK] key "OFF".                    */
2510 '*********************************************************/
2520 POKE &H417,0
2530 RETURN
```

APPENDIX C

This Appendix contains a description of SYMPTOM.EXE, compiled in Turbo C++, Version 1.0 (12), for a Z-248 (IBM 286 compatible machine) using MS-DOS Version 3.0. This program was used to generate output files in ASCII format, which were later used to produce graphs (not shown in text).
Purpose: The purpose of SYMPTOM.EXE is to process data files written by the SYMPTOM.BAS program.

Overview of SYMPTOM.BAS: SYMPTOM.BAS administers a questionnaire detailing possible symptoms that a subject may be experiencing. The questions are read by the program at run-time, and are randomly indexed by question number. SYMPTOM.BAS writes a data file for each session, and the form of this file is:

File Name: nnnTss.SYM

nnn -> Three digit subject number
T -> Test week
ss -> One or two digit session number

First Line: Time/Date Stamp

Subsequent Lines: qq rrr

qq -> Two digit question number
rr -> One, two or three digit response

Overview of SYMPTOM.EXE: SYMPTOM.EXE accepts command-line arguments detailing the input path and file specification, the output path, the number of questions to be expected. A fourth switch selects whether the processed file will be stripped of question and session numbers (a raw file).

To invoke SYMPTOM.EXE the command line is as follows (optional parameters are in angle brackets <optional>, case is not important):

SYMPTOM Fpath\file-specification <O>path Qnumber of questions <R>

Spaces must not be present between the command-line switches and their arguments. Only the <O> and <R> switches are optional. All others MUST be specified. Input file-specification is usually of the form '*.SYM' to get all files on the input path with the extension '.SYM'.

SYMPTOM.EXE reads the input path's directory, looking for matches to the file-specification. If it finds no matches, it will abort with a message. Otherwise the program will sort the directory by SESSION NUMBER and begin reading/processing subject data files. SYMPTOM.EXE will print the name of each data file as it is read.

A listing of the program follows:

```
/*****************************************************************
 **      NAVAL AEROSPACE MEDICAL RESEARCH LABORATORY.            **
 **                                                              **
 **      DEPARTMENT: Medical Research.                           **
 **                                                              **
 **      DIVISION:   Aviation Performance                        **
 **                                                              **
 **      AUTHOR(S):  Andrew H. McCardie                          **
 **                                                              **
 **      PROGRAMMER: Andrew H. McCardie                          **
 **                  Aviation Performance                        **
 **                                                              **
 **      DATE:   29 January 1992                                 **
 **                                                              **
 **      PROJECT:  Sustained Operations 6.2 and 6.3              **
 **                                                              **
 *****************************************************************
 **                                                              **
 ** This program is designed to read raw data files from        **
 ** SYMPTOM.BAS                                                 **
 ** and generate matrices containing the subject's responses.   **
 **                                                              **
 ** Each question number has its own column, while rows         **
 ** represent session numbers.                                  **
 **                                                              **
 ** Matrix files are saved as ###.MAT or as ###R.MAT            **
 ** depending on whether or not the R command line option       **
 ** was specified for RAW data generation.  This option will    **
 ** supress all row and column titles.                          **
 **                                                              **
 ** Usage for SYMPTOM.EXE is:                                   **
 **                                                              **
 ** SYMPTOM F<FULL PATH WITH FILESPEC> Q<NUMBER OF QUESTIONS>   **
 **         O<PATH> R                                           **
 **                                                              **
 ** Ex.: My data is on a floppy in drive B: in a subdirectory   **
 **      called DATA.  The data files are named ###T###.SYM.    **
 **      I want a formatted matrix output, with the data        **
 **      saved in C:\MYDIR.  There are 29 questions.            **
 **                                                              **
 **      The proper command is                                   **
 **                                                              **
 **          SYMPTOM FB:\DATA\*.SYM Q29 OC:\MYDIR                **
 **                                                              **
 **      If I had wanted raw data output, I would have typed     **
 **                                                              **
 **          SYMPTOM FB:\DATA\*.SYM Q29 OC:\MYDIR R              **
 **                                                              **
 **      The 'O' and the 'R' switches are OPTIONAL.  If the      **
 **      'O' switch is not used, data files will be saved in     **
 **      the currently logged directory.                         **
 **                                                              **
 *****************************************************************
 **                                                              **
 **            Last Revision:  January 29, 1992                  **
 **                                                              **
 *****************************************************************/
```

```
#include <stdio.h>
#include <stdlib.h>
#include <dir.h>
#include <dos.h>
#include <conio.h>
#include <errno.h>
#include <string.h>
extern int errno;

int get_dir(char *filespec, char **names)
{
// This function reads the directory of files from the requested
// path and stores them in the array names[]

    struct ffblk myfileblk;
    int result, i,j, index = 0;

    result = findfirst(filespec, &myfileblk,0);
    if (errno == ENOENT || errno == EINVAL)
    {
        printf("No filespec matching <%s> found!\n", filespec);
        exit(1);
    }
    while (!result)
    {
        names[index] = (char *)malloc(13*sizeof(char));
        // Gets 13 bytes

        if (names[index] == NULL)
        {
            printf("Out of memory in 'get_dir'\n");
            exit(1);
        }

        strcpy (names[index], myfileblk.ff_name);
        result = findnext(&myfileblk);

        if (strchr(names[index], 'T') != NULL) index++;
        // Only TEST week will be read.

    }
    return (index); // How many filenames were read?
}

int find_rank (char *test)
{
    /*
        This function returns the integer value of the session
        number given the data file name.
    */
    int i;

    i = strlen(test);
    while (test[i] != '.') i--;
    i--;
    while (isdigit(test[i])) i--;
    return atoi(test+i+1);
}
```

```
void sort_dir (char **names, int index)
{
    /*
        This function performs a bubble sort of directory entries
        given a character array and the length of the array.
        Pointers were used for speed.  This function sorts on
        the actual session number and not the total file name.
    */
    int i,j;
    char *temp;

    for (i=0; i<index-1; i++)
    {
        for (j=i; j<index; j++)
        {
            if (find_rank(names[i]) > find_rank(names[j]))
            {
                temp = names[i];
                names[i] = names[j];
                names[j] = temp;
            }
        }
    }
}

void sort_items ( char **items, int length )
{
    /*
        This function sorts subject responses by question number.
        This is a standard bubble sort accepting a pointer to any
        character array and the length of that array.
    */

    int i,j;
    char *temp;

    for (i=0; i<length-1; i++)
    {
        for (j=i+1; j<length; j++)
        {
            if (atoi(items[i]) > atoi(items[j]))
            {
                temp = items[i];
                items[i] = items[j];
                items[j] = temp;
            }
        }
    }
}

int pad_items (char **items, int length, int number_of_questions)
{
    /*
    This function was added to compensate for missing questions.
    An early version of SYMPTOM.BAS left out one question at
    random for each administration.  This function is called
    whenever the expected number of questions is greater than the
    actual number of questions that were read.
    */
    int i, j, newlength = length;
    char temp[16];
```

```
    j = 0;
    printf("Missing data: ");

    for (i=0; i<=length; i++)
    {
        while ((atoi(items[i]) != j)
               &&(j < number_of_questions))
        {
            printf("[%d]",j+1);
            items[newlength] = (char *)malloc(16*sizeof(char));
               // Adds an array element to the end of items[]
               // for the next missing item.
            if (items[newlength] == NULL)
            {
                printf("Out of memory in 'pad_items'\n");
                exit(1);
            }
            sprintf (temp, "%d  %d", j, -99);
            // -99 is the missing data value
            strcpy (items[newlength], temp);
            newlength++;
            j++;
        }
        j++;
    }
    return newlength;
}

int read_file (char *drive, char *dir, char *name, char **items,
               int number_of_questions)
{
    FILE *FP;
    int i, length;
    char full_name[128], temp[256];
    length = 0;
    strcpy(full_name, drive);
    strcat(full_name, dir);
    strcat(full_name, name);

    if ((FP = fopen (full_name, "r")) == NULL)
    {
        printf("\n*** ERROR READING FILE ***\n");
        printf("%s\n",name);
        exit(1);
    }
    else
    {
        fgets (temp,255,FP);        // Read the TIME/DATE stamp

        do
        {
            fgets (temp,255,FP);    // This code reads the data
                                    // file one line at a time.
            items[length] =
               (char *)malloc((strlen(temp)+1)*sizeof(char));
            // Allocates memory for the next directory entry in
            // items[].
            if (items[length] == NULL)
            {
                printf("Out of memory in 'read_file'\n");
                exit(1);
```

```
            }
            strcpy(items[length], temp);
            if (!feof(FP)) length++;
        } while (!feof(FP) && length <= 30);
        // Won't read over 30 lines
        // in a data file.
        free(items[length]);
        fclose (FP);
    }
    clreol();

    /*
    If I read fewer questions than
    expected, process the missing data.
    Missing data have a '-99' stored
    in their respective fields.
    */
    if (length < number_of_questions)
    {
        sort_items (items, length);
        length = pad_items (items, length, number_of_questions);
    }
    // If too many questions were read,
    // print the error message and exit.
    else if (length > number_of_questions)
    {
        printf("ERROR:  Found %d too many questions\n",
                length-number_of_questions);
        exit(1);
    }
    sort_items (items, length);

    return length;
}

int process_line (char **items, int length, char *line_out)
{

    // The buffer line_out is used to hold one formatted line of
    // output to be written to the workfile.

    int i, j;
    static int count = 0;
    char temp[8];

    strcpy (line_out, "\0");
    /*
    This code moves index j
    from the end of the
    response string (question #)
    to the start of the
    subject's actual response
    and copies the response to
    line_out.
    */
    for (i=0; i<length; i++)
    {
        j = strlen(items[i]) - 1;
        while (isspace(items[i][j])) j--;
        while (!isspace(items[i][j])) j--;
        sprintf(temp, "%4d", atoi(items[i]+j));
        strcat(line_out, temp);
```

```
    }

    strcat(line_out, "\n");

    count++;                    // How many sessions have I
    return count;               // read?
}

int parse_command (int argc, char *argv[], char *filespec,
                   int *number_of_questions, char *extension,
                   char *outpath)
{
    // This function processes command-line input and returns
    // filespec, number_of_questions, and extension.

    int i, flag = 0, raw = 0;

    i = argc-1;
    while (i > 0)
    {
        switch (argv[i][0])
        {
            case 'F' :
            case 'f' : strcpy (filespec, &argv[i][1]);
                       flag++;
                       break;
            case 'O' :
            case 'o' : strcpy (outpath, &argv[i][1]);
                       if ((outpath[strlen(outpath)-1] != '\\')
                          &&(outpath[strlen(outpath)-1] != ':'))
                          {
                             strcat(outpath, "\\");
                          }
                       break;
            case 'Q' :
            case 'q' : *number_of_questions = atoi(&argv[i][1]);
                       flag++;
                       break;
            case 'R' :
            case 'r' : raw = 1;
                       strcpy(extension, "R.MAT");
                       printf("Writing raw data file...\n");
                       break;

            default  : printf("WARNING: %s is not"
                              " a valid switch\n", argv[i]);
                       delay (1000);

        }
        i--;
    }

    if (flag < 2)
    {
        printf("\n\nUSAGE: SYMPTOM F<PATH\FILESPEC> Q<NUMBER"
               " OF QUESTIONS> ");
        printf("O<OUTPUT PATH> R \n\n");
        printf("The FILESPEC should contain wildcards unless\n");
        printf(" a single file is to be analyzed.\n\n");
        printf("The <O> and <R> parameters are optional, and"
               " select an\n");
        printf(" <O>utput path and RAW data output"
```

```
                "respectively.\n");
        exit(1);
    }
    return raw;
}


void main (int argc, char *argv[])
{
    FILE *FP;
    int index, i, j, temp, raw, length, number_of_questions;
    // Setting raw = 1 will cause unformatted
    // data to be saved to the workfile.

    // names[] contains the list of files to be read
    // items[] contains the actual response data
    char **names, **items;


    char filespec[80], dir[40], drive[3], workfile[20],
         line_out[132], extension[8], outpath[32];

    index = 0;
    strcpy(outpath, "\0");
    names = (char **)malloc(256*sizeof(char *));
     // names[] can contain 256 directory entries.
    items = (char **)malloc(256*sizeof(char *));
     // items[] can contain 256 questions.
    strcpy (extension, ".MAT");
     // Copies the default workfile extension into the extension
     // variable.

    raw = parse_command (argc, argv, filespec,
                         &number_of_questions, extension,
                         outpath);

    fnsplit (filespec, drive, dir, NULL, NULL);

    clrscr();

    printf("Reading directory...\n");

    index = get_dir(filespec, names); // Reads the disk directory
                                      // and stores the requested
                                      // file names in names[]
    sort_dir(names, index);
    sprintf(workfile, "%s%d%s", outpath, atoi(names[0]),
            extension);
                                      // Names the workfile as
                                      // ###.MAT or ###R.MAT
                                      // where ### is the subject
                                      // number and R specifies
                                      // an RAW data file.


    if ((FP = fopen (workfile, "a")) == NULL)
    {
        printf("\n*** ERROR WRITING OUTPUT FILE ***\n");
        printf("%s\n",workfile);
        exit(1);
    }
    else
```

```
{
    for (i=0; i<index; i++)
    {
        gotoxy(1,3);
        printf("Reading file... <%s>\n", names[i]);
        length = read_file(drive, dir, names[i],
                           items, number_of_questions);
                               // This function reads in all
                               // subject answers for the
                               // data file in names[i]

        temp = find_rank(names[i]);
                               // This code assigns
                               // the actual session number
                               // from the current data file
                               // to the name of the
                               // workfile.

        if (!raw)
        {
            if (process_line(items, length, line_out) == 1)
            {
                // This code writes the
                // question numbers across
                // the top of the workfile.
                fprintf(FP, "   | ");
                for (j=0; j<length; j++)
                    fprintf(FP, "%3d|", j+1);
                fprintf(FP, "\n");

                // This code inserts the
                // correct number of equal
                // signs in the workfile.
                for (j=0; j<length*4+5; j++)
                    fprintf(FP,"=");
                fprintf(FP, "\n");
            }
            fprintf(FP, "%3d|%s", temp, line_out);
        }
        else
        {
            // Print unformatted
            // output to workfile.
            process_line(items, length, line_out);
            fprintf(FP, "%s", line_out);
        }
    }
    fprintf(FP, "\n");
fclose(FP);
}
}
```

APPENDIX D

This appendix contains the instructions that are displayed to the subjects at the beginning of each session of the drug symptom questionnaire.

INSTRUCTIONS

In this questionnaire, you are asked to rate the degree or intensity of a set of symptoms, feelings, or behaviors. Each symptom will be presented individually on the screen above a line segment scale measuring from 0 to 100. The arrow can be moved on the 0-100 scale to indicate the intensity of that particular symptom.

Press the blue or red keys on the modulus box until the position of the arrow is most appropriate to the intensity of the symptom as you are experiencing it NOW.

The 0 means no sensation of that symptom whatsoever, and the 100 means as intense as you could imagine experiencing it.

After you have made your judgement of the symptom intensity, record your response by pressing the tapper key on the modulus box.

YOU MUST RESPOND TO EACH ITEM.